IJISEM
INTERNATIONAL JOURNAL OF INNOVATIONS IN SCIENCE ENGINEERING AND MANAGEMENT

International Journal of Innovations In Science Engineering And Management

OPEN ACCESS

Volume: 5

Issue: 3

Month: August

Year: 2026

ISSN: 2583-7117

Published: 22.08.2026

Citation:

Amit Kumar Thakore, Neha Gupta, Akash Saxena, Amar Singh Verma "Generative Adversarial Networks for Anomaly and Malware Detection" International Journal of Innovations in Science Engineering and Management, vol.5 , no.3 , 2026, pp. 416- 425

DOI:

10.69968/ijisem.2026v5i3416-425

Generative Adversarial Networks for Anomaly and Malware Detection

Amit Kumar Thakore[1], Neha Gupta[2], Akash Saxena[3], Amar Singh Verma[4]

[1]*Computer Science and Engineering Dr. K. N. Modi University India*
[2]*Computer Science and Engineering Dr. K. N. Modi University India*
[3]*Computer Science and Engineering CITM, Jaipur India*
[4]*Computer Science and Engineering Dr. K. N. Modi University* India

Abstract

Generative Adversarial Networks or GANs, have become a significant approach in deep learning along with Con-volutional and Recurrent Neural Networks, due to improvements in computing technology and more advanced ways to train these frameworks or models. Since GANs were first introduced in 2014, their application has expanded beyond image generation to include critical security tasks like anomaly detection and malware analysis. This paper offers a comprehensive survey of how GAN-based methods are utilized for identifying unusual and harmful activities in cyber settings. It examines key variants of GANs relevant to this field, explains their fundamental architectures and training methods, and explains their integration into systems to detect anomalies and malware. Additionally, the paper catalogs publicly accessible datasets and evaluation metrics frequently used in the reviewed studies to illustrate common experimental methodologies and research directions. Finally, it addresses ongoing challenges and potential future avenues for employing GANs to counteract emerging cybersecurity threats, highlighting their importance in developing more proactive and robust security measures.

Keywords; Generative Adversarial Networks (GANs), mal-ware detection, cyber setting, deep neural networks.

INTRODUCTION

Generative Adversarial Networks (GANs) form a class of deep learning models designed to learn and sample from complex data distributions through adversarial training be-tween a generator and a discriminator. In their standard form, GANs are widely used to train generative models such as deep convolutional neural networks that can synthesize realistic data, including images and high-dimensional feature representations. By learning from existing samples, GAN-based models are able to produce previously unseen but statistically consistent instances, which makes them suitable for augmenting and balancing datasets in security applications. In cybersecurity, this capability is increasingly exploited to construct synthetic corpora that resemble real-world attack traffic or malware behaviors, thereby supporting the design and evaluation of robust defense mechanisms [1].

Malware, or malicious software, refers to programs in-tentionally crafted to disrupt normal system operation, gain unauthorized access, or exfiltrate sensitive information. The shift to large-scale remote work during and after the COVID- 19 pandemic has coincided with a substantial growth in cybercrime, including large volumes of malware activity. Re-ports from the industry reveal that in 2021 and 2022, there were billions of recorded malware attacks, with hundreds of thousands of new samples identified each day. The total number of unique malware programs has exceeded one billion, highlighting the extensive and ongoing nature of this threat. Simultaneously, attackers are increasingly utilizing machine learning to automate and refine their strategies, resulting in advanced and quickly changing threats that pose challenges to conventional detection systems.

In this scenario, Generative Adversarial Networks (GANs) present a way to reveal ”hid-den” malicious activities by creating realistic adversarial or rare samples, which can strengthen learning-based detectors against new attack variations [2],[4].

This survey explores a wide range of research that utilizes GAN architectures for malware detection and characterization, focusing on their contribution to enhancing network and sys-tem security. Malware is frequently viewed as a versatile arse-nal for attackers, capable of executing various actions such as disabling services, elevating privileges, or extracting sensitive information without user permission. Within this environment, GANs serve as a double-edged sword: on one hand, attack frameworks like MalGAN illustrate that adversarially trained generators can create malware variants that evade black-box static detection systems with nearly zero detection rates; on the other hand, defensive research employs GANs to produce challenging synthetic samples, thereby boosting the resilience of intrusion and malware detectors through adversarial training and data enhancement. This paper analyzes both sides, doc-umenting how GAN-based models can be exploited to avoid classifiers while also detailing how they can be incorporated into defense strategies to improve the detection of obfuscated and zero-day threats.

Building on previous surveys of GAN types, this review categorizes representative models based on their application areas, such as mobile networks, traffic analysis, Internet of Things settings, physical-layer security, and broader cyber-security applications. The discussion pays special attention to persistent technical issues, including unstable training dy-namics, convergence behavior, and mode collapse, along with suggested solutions involving enhanced objective functions and regularization techniques. By linking GAN architectures to their specific applications and summarizing their advantages and drawbacks, this work provides a comprehensive overview of how various GAN families contribute to security-related efforts in computer and communication networks. The primary objective of this review is to deliver a clear, practical under-standing of GAN applications in cybersecurity, particularly in malware and threat-focused research. For researchers and prac-titioners, the survey serves as a reference point for selecting suitable GAN variants, identifying appropriate datasets and evaluation strategies, and recognizing open problems where GAN-based techniques can be further exploited to counter emerging cyber threats.

Overview of GAN

A Generative Adversarial Network (GAN) consists of two neural networks trained jointly in an adversarial setting: a generator and a discriminator. The generator receives random noise as input and synthesizes artificial samples intended to resemble data drawn from the true distribution, while the discriminator acts as a binary classifier that distinguishes between real samples and those produced by the generator.

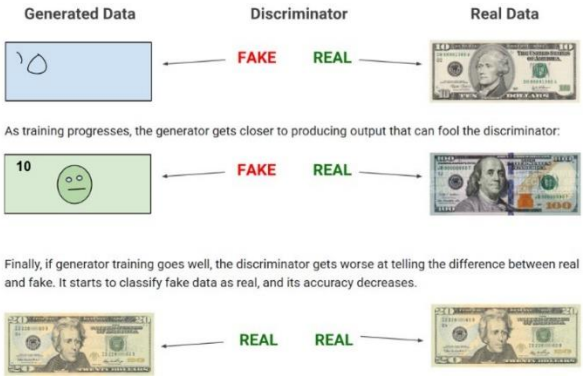

Figure 1. An image depicting the entire system

At the start of training, the generator outputs clearly unre-alistic samples, and the discriminator can easily classify them as fake.

As training progresses, both networks are updated via backpropagation: the discriminator adjusts its parameters to improve its real–fake classification accuracy, and the generator updates its weights based on the discriminator’s feedback so that its outputs become progressively more realistic. In a successful training run, this iterative process converges toward a point where the discriminator can no longer reliably separate generated samples from genuine data, indicating that the generator has learned a high-fidelity approximation of the target data distribution.

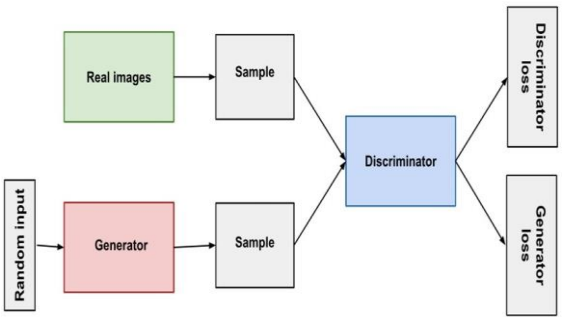

Figure 2. Generator and Discriminator

Technical Terms and Definitions

A Generative Adversarial Network (GAN) is a machine learning framework composed of two distinct deep neural net-works that are trained simultaneously in an adversarial setting. One network, the generator, receives random noise as input and learns to produce samples that mimic the characteristics of real data, while the second network, the discriminator, evaluates both genuine and generated samples and classifies them as real or fake.

Training is driven by backpropagation over a minimax objective: if the discriminator correctly labels a sample, its parameters are updated to reinforce this decision, and the generator receives a gradient signal that pushes it to reduce the discriminator's ability to distinguish its outputs from real data. Conversely, when the discriminator misclassifies a sample, its parameters are adjusted to improve future discrimination. This iterative process continues until the discriminator's accuracy approaches that of random guessing, indicating that the gener-ator has learned to produce samples that are statistically close to the true data distribution.

In this setting, the generator's role is to approximate the un-derlying data distribution and produce realistic synthetic data, while the discriminator's role is to estimate the probability that a given input originates from the real dataset rather than from the generator. GANs were originally introduced as a type of generative model under the umbrella of unsupervised learning, but subsequent work has demonstrated their applicability to semi-supervised, supervised, and reinforcement learning sce-narios through suitable modifications of the loss functions and training protocols.

Working principle of GAN

A standard GAN architecture, as illustrated conceptually in Fig. 1, consists of a generator–discriminator pair that forms a two-player zero-sum game, where the generator aims to minimize the discriminator's ability to distinguish real and fake data, and the discriminator aims to maximize this dis-crimination performance [8]-[10]. In its basic formulation, the model is trained on samples from a target dataset, and the generator learns to map points from a latent noise space to the data space while the discriminator learns a decision boundary between real and generated samples.

Generator: The generator network is responsible for mapping noise vectors to synthetic samples that approximate the real data distribution [12]. It learns to capture the latent structure of the training data so that, over time, its outputs become visually or statistically indistinguishable from genuine examples. During training, the generator is updated to "fool" the discriminator, i.e., to increase the discriminator's probability of labeling generated samples as real.

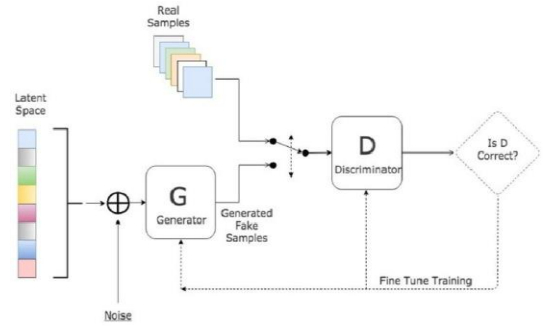

Figure 3. Generative Adversarial Network

Discriminator: The discriminator functions as a binary classifier that is trained on a combination of genuine samples from the dataset and artificial samples generated by the generator [13]. Its goal is to accurately determine the origin of each input, which in turn provides a continuous learning signal that informs the generator. As the training advances, the discriminator enhances its internal understanding to better distinguish between real and synthetic data, while the generator adjusts its strategies to take advantage of any flaws in this understanding.

Feedback loop: The relationship between the generator and the discriminator forms a closed feedback loop that is cru-cial for optimizing Generative Adversarial Networks (GANs). The gradients obtained from the loss associated with the discriminator's classifications are not only sent back through the discriminator but are also used to update the generator, allowing both networks to be adjusted simultaneously. While this adversarial system is an effective approach for learning intricate distributions, it also presents challenges such as unstable training behaviors and mode collapse. These issues

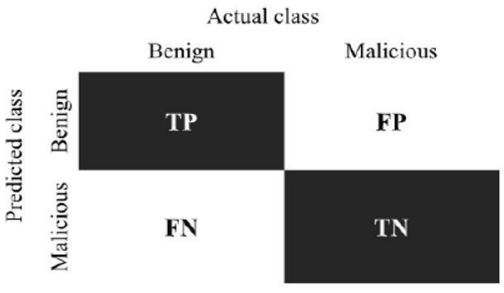

Figure 4. Machine learning Classifications Performance Metrics

Measuring performance

In evaluating the performance of the proposed machine learning models, standard classification metrics derived from the confusion matrix are utilized. These metrics include accu-racy, precision, recall, F1-score, along with the tallies of true positives, true negatives, false positives, and false negatives, as shown in the accompanying figure.

In this context, a true positive (TP) denotes the number of benign samples correctly classified as benign, whereas a true negative (TN) denotes the number of malicious samples correctly identified as malicious. A false positive (FP) corre-sponds to malicious samples incorrectly labeled as benign, and a false negative (FN) represents benign samples incorrectly labeled as malicious. The overall classification accuracy is computed as the proportion of correctly classified test samples to the total number of test samples, i.e.,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision quantifies the reliability of positive predictions and is given by

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

while recall (also referred to as the true positive rate or sensitivity) measures the fraction of actual positive instances that are correctly detected,

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

The F1-score, defined as the harmonic mean of precision and recall, summarizes the trade-off between these two metrics,

$$F_1 \text{ Score} = \frac{2\times(\text{Precision}\times\text{Recall})}{\text{Precision}+\text{Recall}} \quad (4)$$

This family of metrics is widely adopted in evaluating classification models for security and generative tasks and has been shown to correlate well with qualitative assessments in various benchmark datasets.

Datasets

To capture different aspects of anomaly and malware-related behavior, the surveyed studies rely on a diverse set of benchmark datasets. Among these, Channel State Information (CSI) data for wireless sensing and the Microsoft Malware Classification Challenge dataset play a central role in shaping the reported results, complemented by network traffic, RF signal, and image-based malware corpora.

WiFi RSSI:: The Wi-Fi Received Signal Strength Indicator (RSSI) datasets are designed for indoor localization and user detection based on signal strength fingerprints collected from multiple access points [27]. They support applications such as smart home automation, indoor security monitoring, and occupancy estimation, by enabling models that map RSSI vectors to user positions or regions within a building.

CSI:: Channel State Information datasets provide fine-grained measurements of the wireless channel, which are widely used for human sensing tasks such as activity recogni-tion, person identification, and people counting. Recent CSI datasets, for example those collected over 80 MHz IEEE 802.11ac links, record variations induced by human motion across diverse environments, thereby enabling the development and evaluation of machine learning algorithms for behavior recognition and domain-adaptive sensing.

Network traffic (KDD Cup '99):: The KDD Cup 1999 dataset is a classic intrusion detection benchmark composed of millions of network connection records, each described by 41 features and labeled as normal or as one of several attack types (e.g., DoS, probe, user-to-root, remote-to-local). Despite known limitations and redundancy, it remains a widely used reference for training and evaluating intrusion detection models and for comparative assessment of new techniques.

DeepSig RadioML 2016.10A:: The RadioML 2016.10A dataset, released by DeepSig, is a synthetic RF dataset gener-ated with GNU Radio and comprising 11 modulation schemes (8 digital and 3 analog) over a range of signal-to-noise ratios. It is primarily used for research on automatic modulation classification and RF signal processing, providing labeled IQ samples for evaluating deep learning-based communication and sensing models.

Microsoft Malware Classification Challenge:: The Microsoft Malware Classification Challenge dataset is a large-scale benchmark of approximately 20K Windows PE malware samples, provided as raw hexadecimal byte dumps and corre-sponding disassembly reports [32]. Once uncompressed, the dataset is nearly 0.5 TB in size and covers nine malware families, with each sample labeled by a 20-character identifier and an integer family index; this dual representation supports feature extraction from both byte-

level and disassembly-level views and has become a standard testbed for malware family classification and related tasks.

Malimg:: The Malimg dataset consists of 9,339 grayscale images generated from malware binaries belonging to 25 different families. Each binary is transformed into an image (e.g., byteplot) so that visual patterns characteristic of different families can be captured and exploited by image processing and deep learning methods. Malimg is widely used for image-based malware detection and classification and often serves as a complement to PE- or opcode-level datasets in experimental studies.

TYPES OF GAN

Conditional GAN (CGAN)

Conditional Generative Adversarial Networks (CGANs) ex-tend the original GAN framework by conditioning both the generator and discriminator on auxiliary information, such as class labels or attributes. In a typical CGAN, the generator re-ceives a noise vector concatenated with a condition (e.g., label or feature vector), and the discriminator evaluates whether an input sample is real or generated given the same condition. This design enables controlled generation of samples with specified properties and is widely used for tasks such as attribute-specific image synthesis, text-to-image generation, and tabular data augmentation.

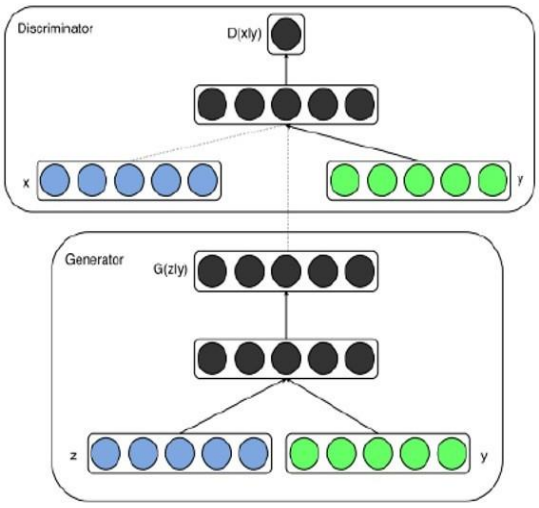

Figure 5. Architecture of Conditional GAN

Wasserstein GAN (WGAN) and WGAN-GP

Wasserstein GANs replace the original Jensen–Shannon divergence–based objective with an approximation of the Wasserstein (Earth Mover's) distance between the real and generated distributions, leading to more stable training and a loss that better correlates with sample quality. To enforce the required Lipschitz constraint, the discriminator (often called the critic) initially uses weight clipping, which is later im-proved by WGAN with Gradient Penalty (WGAN-GP), where a gradient norm penalty is applied instead of hard clipping. These variants generally produce higher-quality samples and reduce issues such as mode collapse compared with vanilla GANs.

RNN-based GANs

RNN-based GANs employ recurrent neural networks in the generator, discriminator, or both, to handle sequential data such as text, time series, or logs. Architectures such as LSTM-GAN incorporate Long Short-Term Memory units to capture long-range temporal dependencies, making them suitable for generating longer sequences with complex dynamics, while simpler RNN-GANs can be effective for shorter sequences or scenarios with limited context. In these models, the generator outputs sequences token by token (or step by step), and the discriminator evaluates whole sequences, enabling adversarial learning directly in sequence space.

Deep Convolutional GAN (DCGAN)

Deep Convolutional GANs (DCGANs), introduced by Rad-ford et al., popularized a set of architectural guidelines for stable image generation using convolutional neural networks. DCGANs replace fully connected layers with strided convolu-tions and transposed convolutions, adopt batch normalization, and use ReLU activations in intermediate generator layers with tanh or sigmoid at the output to constrain pixel ranges. On the discriminator side, leaky ReLU activations and convolutional downsampling are employed to improve gradient flow and representation learning. DCGAN has become a foundational architecture for many subsequent image-based GAN variants.

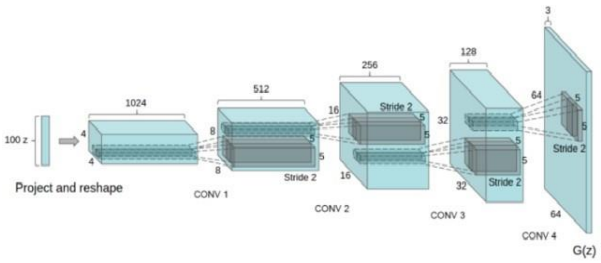

Figure 6. The Architecture of DCGANs

InfoGAN

InfoGAN is an information-theoretic extension of GANs that maximizes the

mutual information between a subset of latent variables and the generated observations. The latent vec-tor is decomposed into incompressible noise and "structured" codes, and an auxiliary network is introduced to approximate the posterior over these codes, enabling an additional mutual-information regularization term in the objective[36]. This encourages the model to learn disentangled and interpretable latent factors, which has been demonstrated on datasets such as MNIST, CelebA, and SVHN, where variations in latent codes correspond to meaningful changes in digit style, facial attributes, or object appearance.

CycleGAN

CycleGAN [38] is an image-to-image translation framework that learns mappings between two visual domains using unpaired training data. It employs two generators (one for each direction) and two discriminators, together with a cycle-consistency loss that enforces that translating an image from one domain to the other and back reconstructs the original input. This design enables translation tasks such as style trans-fer, domain adaptation, and object appearance modification without requiring aligned image pairs, extending earlier paired approaches such as Pix2Pix [39].

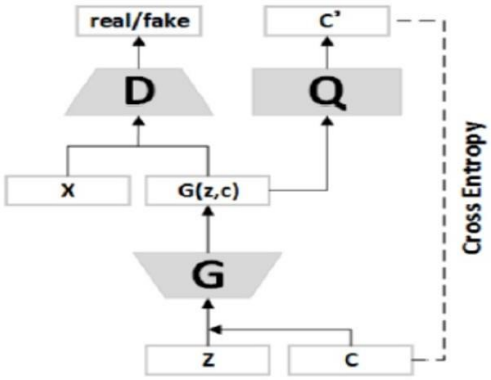

Figure 7 The Architecture of InfoGAN

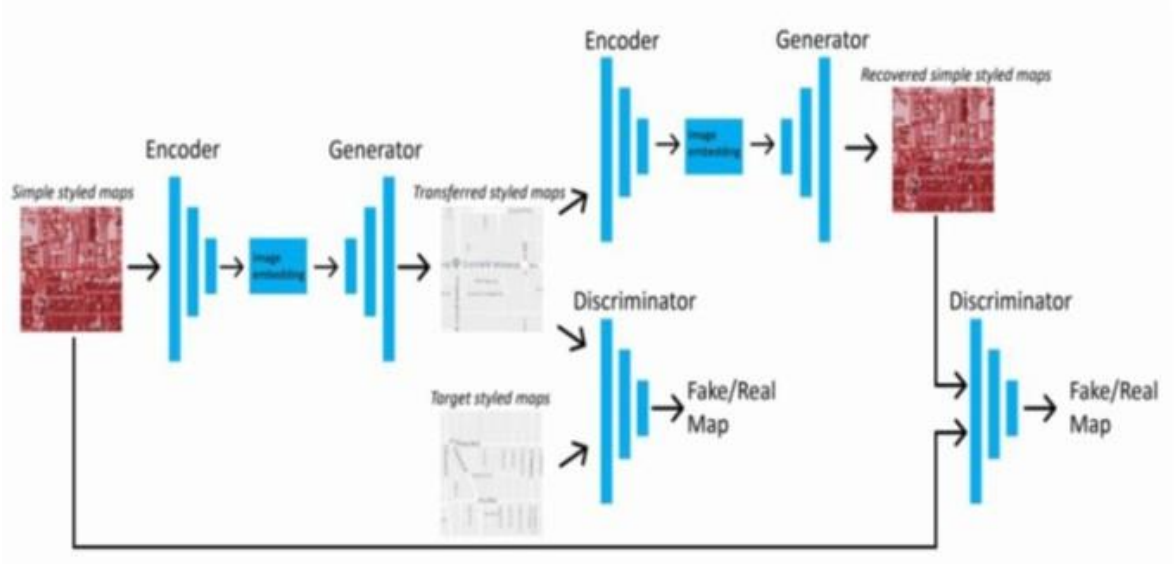

Figure 8. The Architecture of CycleGAN

Auxiliary Classifier GAN (AC-GAN)

Auxiliary Classifier GAN (AC-GAN) augments the discrim-inator with an additional classification head that predicts class labels for both real and generated samples. The generator is conditioned on class information and is trained to produce samples that are both realistic and class-discriminative, while the discriminator jointly optimizes an adversarial loss and a supervised classification loss [17][41]. Compared with CGAN, which conditions implicitly through concatenation, AC-GAN explicitly structures the latent space via the auxiliary classifier, often improving class-conditional image quality and label controllability.

- **Bi-directional GAN (BiGAN)**

BiGAN enhances the tra-ditional GAN framework by incorporating an encoder network that translates data samples back into the latent space, creating a system composed of three networks: the generator, the discriminator, and the encoder [42]. The discriminator's role is to differentiate between pairs of (latent, real data) and (latent, generated data), which encourages the encoder and

generator to act as approximate inverses of each other. This bidirectional setup allows for representation learning and inference within the latent space, aiding in various tasks such as feature extraction, anomaly detection, and semi-supervised learning.

- **Temporal GAN (TGAN)**

Temporal Generative Adversarial Networks (TGANs) focus on video and temporal datasets by explicitly addressing the time dimension [44]. A standard TGAN features a temporal generator that generates a sequence of latent codes corre-sponding to video frames, along with an image generator that converts each latent code into a frame. This design enables the model to effectively capture both temporal dy-namics and spatial characteristics. Numerous TGAN variants utilize Wasserstein-style objectives and end-to-end training approaches to stabilize the learning process and have shown the capability to create diverse and coherent video sequences from unlabeled data.

- **Least Squares GAN (LSGAN)**

Least Squares GANs modify the discriminator's loss from a sigmoid cross-entropy objective to a least-squares loss, which helps mitigate vanishing gradients and encourages generated samples to move closer to the decision boundary in a more controlled manner [44]. This adjustment often yields smoother gradients for the generator, leading to higher-quality images and more stable training behavior. LSGANs have been suc-cessfully applied in unsupervised and conditional generation tasks as an alternative to the original GAN loss.

- **Evolutionary GAN (E-GAN)**

Evolutionary Generative Adversarial Networks (E-GANs) incorporate evolutionary computation concepts into GAN training by maintaining a population of generators G that evolve over time. At each iteration, variation operators (e.g., mutation strategies that modify objective components or hy-perparameters) generate offspring generators, which are then evaluated using a fitness function derived from the discrim-inator's feedback. Generators with high fitness are chosen to create the subsequent generation, while the discriminator D is continuously refined to differentiate real samples from generated ones. This framework seeks to enhance robustness and mitigate challenges like mode collapse by exploring a variety of generator strategies.

- **Boundary Equilibrium GAN (BEGAN)**

Boundary Equilibrium GAN (BEGAN) [47] employs an autoencoder-based discriminator and introduces an explicit equilibrium mechanism to balance generator and discriminator capacities. The discriminator is trained to minimize recon-struction loss on real samples and maximize it for generated samples, while the generator aims to produce samples that minimize the discriminator's reconstruction error. A control variable

MS Word me **Alt** + = press karein aur in equations ko paste karke **Enter** dabayein:

$$L_D = L(x) - k_t \cdot L\big(G(z_D)\big) \quad , \quad \text{for } \vartheta_D$$

$$L_G = L\big(G(z_G)\big) \quad , \quad \text{for } \vartheta_G$$

$$k_{t+1} = k_t + \lambda_k \left(\gamma L(x) - L\big(G(z_G)\big)\right) \quad \text{for each training step } t$$

is updated at each step to maintain a target ratio between real and generated reconstruction losses, promoting stable training and high-quality image synthesis.

Progressive Growing GAN (ProGAN)

Progressive Growing of GANs (ProGAN), proposed by Kar-ras et al., trains GANs by gradually increasing the resolution of generated images and discriminated inputs. Training begins at a low spatial resolution, where the generator and discriminator learn coarse structure, and new layers are progressively added to both networks to handle higher resolutions, with smooth transition phases to avoid instability. Combined with improved loss functions such as WGAN-GP and LSGAN, this strategy enables the generation of high-resolution, photorealistic im-ages and has influenced many subsequent large-scale genera-tive models.

MSG-GAN

Multi-Scale Gradient GAN (MSG-GAN) addresses gradient flow and stability issues in high-resolution generation by allowing gradients from the discriminator to be propagated to the generator at multiple image scales. Instead of only feeding the final high-resolution output to the discriminator, intermediate generator feature maps at different resolutions are also connected to corresponding discriminator branches. This multi-scale feedback improves signal propagation, reduces noisy gradient effects, and yields sharper, more consistent high-resolution images, providing an alternative to purely progressive growing schemes.

Types of GAN-based detections

Anomaly Detection

GAN-based anomaly detection has attracted growing atten-tion as attack patterns become more complex and

harder to capture with conventional models. Early work by Navidan et al.[17] provides a broad taxonomy of anomaly types and detection paradigms, motivating the use of advanced learning-based approaches in this area. Wang et al.[46] subsequently introduced AnoGAN, a DCGAN-based framework [13] that trains a generator–discriminator pair exclusively on normal data and identifies anomalies at test time via reconstruction and latent-space consistency measures.

To address the scalability and inversion limitations of AnoGAN, Odena et al.[37] proposed Efficient GAN-Based Anomaly Detection (EGBAD), which leverages the BiGAN [42] architecture to learn an explicit encoder from input space to latent space during adversarial training, thereby reducing inference time and improving anomaly scoring. Donahue et al.[39] later presented GANomaly, a semi-supervised model that combines an encoder–decoder–encoder generator with a discriminator and multiple loss terms, enabling more discrim-inative latent representations of normal data and superior per-formance across several image benchmarks; subsequent work has further refined convergence behavior using techniques such as exponential moving averages and autoencoder-based regularization.

Cyber Intrusion and Malware Detection

In cybersecurity, GANs have been employed both to stress-test and to strengthen intrusion detection systems (IDSs). Several studies demonstrate that GAN-generated adversarial traffic can exploit weaknesses in ML-based IDSs, indicating susceptibility to crafted examples and highlighting the need for robust defenses[16]. For example, GAN-based attacks have been used to learn traffic feature manipulations that cause IDS models to misclassify malicious flows as benign, while follow-up work proposes GAN-assisted data augmentation and hybrid architectures to improve resilience on benchmarks such as UNSW-NB15 and CIC-IDS-2017.

In the malware domain, GANs have been applied across multiple platforms, including Windows and Android, to gen-erate, transform, and detect malicious binaries. Image-based approaches convert malware binaries into grayscale or RGB images and train GANs or DCGAN variants to support detec-tion and family classification, as seen in PlausMal-GAN and related frameworks that address zero-day and class-imbalance challenges. Other work, such as tDCGAN-style models, ex-plores transfer learning and adversarial generation to detect previously unseen malware variants, illustrating the adaptabil-ity of GANs to different operating systems and representation schemes (e.g., PE files, opcode sequences, behavior logs). These contributions collectively show that GANs can be used to perturb traffic, augment datasets, and enhance intrusion and malware detectors.

Security Attacks

Beyond detection, GANs are increasingly utilized to model and generate security attacks for testing and hardening sys-tems. One prominent application is the generation of real-istic but synthetic attack payloads that support vulnerability assessment and automated penetration testing. Chowdhary et al. proposed an autonomous web application penetration test-ing framework that employs semantic tokenization to extract salient features from XSS and SQL injection payloads and uses a conditional sequence GAN to produce new attack strings capable of evading web application firewalls at scale.

By learning the distribution of existing exploits and generat-ing diverse variants, such GAN-based frameworks can system-atically probe web applications and services for weaknesses associated with advanced persistent threats (APTs), input-validation flaws, and other classes of vulnerabilities. These developments highlight the dual function of GANs in the realm of security: they serve as instruments for crafting advanced offensive strategies and as facilitators for more thorough, automated assessments of defensive systems.

CONCLUSION

his study provides a comprehensive survey of Generative Adversarial Networks (GANs) within the context of malware analysis and broader cybersecurity, particularly focusing on tasks related to anomaly and threat detection. By consolidating the fundamentals of GANs, cataloging their variants employed in security contexts, and examining frequently used evaluation metrics, this work offers an integrated framework for under-standing the contributions of GAN-based methods to cyber defense.

The analysis identifies existing research gaps, such as issues with training stability, the lack of standardized benchmarks, and the dual-use potential of GANs, while emphasizing the critical role of precise hyperparameter tuning in enhancing detection accuracy and robustness. Furthermore, the paper suggests promising future directions, including the application of GANs in advanced neural cryptography, more complex attack and traffic representations, and hybrid models that combine GANs with other deep learning and graph-based approaches to improve current cybersecurity solutions.

By presenting a structured overview of GAN applications in malware detection, intrusion detection, and anomaly detection, this survey aims to assist researchers and practitioners in choosing suitable GAN variants, datasets, and design options. In doing so, it seeks to promote a more systematic develop-ment of GAN-enhanced security solutions and encourage fur-ther advancements in resilient network and system protection.

REFERENCES

[1] Bayer et al. "Dynamic Analysis of Malicious Code" vol. 2, pp. 66-67, 2006; Doi: 10.1007/s11416-006-0012-2.
[2] Petrosyan, "Annual number of malware attacks worldwide from 2015 to 2022" 2023. Online, Accessed 2 2 2024.
[3] N. J. Palatty "30+ Malware Statistics You Need To Know In 2024" Astra, 2023. Online, Accessed 2 2 2024.
[4] J. Brownlee, "How to develop an auxiliary classifier GAN (AC-GAN) from scratch with Keras" 2021. Online, Accessed 2 2 2024.
[5] Dunmore et al. "Generative Adversarial Networks for Malware Detection: a Survey" ArXiv, vol. abs/2302.08558, 2023; Doi: 10.48550/arXiv.2302.08558.
[6] S. Gihon, "Ransomware Trends Q4 2023 Report" 2024. Online, Ac-cessed 2 2 2024.
[7] W. Hu and Y. Tan, "Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN" ArXiv, vol. abs/1702.05983, 2017.
[8] Salimans et al. "Improved Techniques for Training GANs" ArXiv, vol. abs/1606.03498, 2016.
[9] Isola et al. "Image-to-Image Translation with Conditional Adver-sarial Networks" in 2017 IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), 2017, pp. 5967-5976; Doi: 10.1109/CVPR.2017.632.
[10] J. Ho and S. Ermon, "Generative Adversarial Imitation Learning" in Neural Information Processing Systems 2016.
[11] Gharakhanian, "Generative Adversarial Networks – Hot Topic in Machine Learning" 2017. Online, Accessed 2 2 2024.
[12] He et al. "Analysis of Image Generation by different Generator in GANs" Journal of Physics: Conference Series, vol. 1903, 2021.
[13] Radford et al. "Unsupervised Representation Learning with Deep Convo-lutional Generative Adversarial Networks" CoRR, vol. abs/1511.06434, 2015.
[14] Goodfellow et al. "Generative Adversarial Nets" in Neural Information Processing Systems, 2014.
[15] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks" ArXiv, vol. abs/1701.04862, 2017.
[16] Cai et al. "Generative Adversarial Networks" ACM Computing Surveys (CSUR), vol. 54, pp. 1-38, 2021.
[17] Navidan et al. "Generative Adversarial Networks (GANs) in Net-working: A Comprehensive Survey & Evaluation" ArXiv, vol. abs/2105.04184, 2021.
[18] Won et al. "PlausMal-GAN: Plausible Malware Training Based on Generative Adversarial Networks for Analogous Zero-Day Malware Detection" IEEE Transactions on Emerging Topics in Computing, vol. 11, pp. 82-94, 2023; Doi: 10.1109/TETC.2022.3170544.
[19] Dutta et al. "Generative Adversarial Networks in Security: A Survey" 2020. 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0399-0405, 2020.
[20] Prabakaran et al. "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders" IET Information Security, vol. 17, no. 3, pp. 315-551, 2023; Doi: 10.1049/ise2.12106.
[21] Salimans et al. "Improved Techniques for Training GANs" ArXiv, vol. abs/1606.03498, 2016.
[22] S. T. Barratt and S. Rishi, "A Note on the Inception Score" ArXiv, vol. abs/1801.01973, 2018.
[23] Borji, "Pros and Cons of GAN Evaluation Measures" ArXiv, vol. abs/1802.03446, 2018.
[24] Dunmore et al. "A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection" IEEE Access, vol. 11, pp. 76071-76094, 2023; Doi: 10.1109/ACCESS.2023.3296707.
[25] Che et al. "Mode Regularized Generative Adversarial Networks" ArXiv, vol. abs/1612.02136, 2017.
[26] Szegedy et al. "Rethinking the Inception Architecture for Computer Vi-sion" in 2016 IEEE Conference on Computer Vision and Pattern Recog-nition (CVPR), 2016, pp. 2818-2826; Doi: 10.1109/CVPR.2016.308.
[27] Rohra et al."User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Al-gorithm with Neural Networks" in International Conference on Soft Computing for Problem Solving, 2016.
[28] Meneghello et al. "A CSI Dataset for Wireless Human Sensing on 80 MHz Wi-Fi Channels" IEEE Communications Magazine, vol. 61, pp. 146-152, 2023.
[29] Yousefi et al. "A Survey on Behavior Recognition Using WiFi Channel State Information" IEEE Communications Magazine, vol. 55, pp. 98-104, 2017.
[30] Ronen et al. "Microsoft Malware Classification Challenge" ArXiv, vol. abs/1802.10135, 2018.

[31] Nataraj et al. “Malware images: visualization and automatic classifica-tion” in Visualization for Computer Security, 2011.
[32] M. Mirza and S. Osindero “Conditional Generative Adversarial Nets” ArXiv, vol. abs/1411.1784, 2014.
[33] Chen et al. “InfoGAN: Interpretable Representation Learning by Infor-mation Maximizing Generative Adversarial Nets” in Neural Information Processing Systems, 2016.
[34] Li et al. “SCGAN: Disentangled Representation Learning by Adding Similarity Constraint on Generative Adversarial Nets” IEEE Access, vol. 7, pp. 147928-147938, 2019.
[35] Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks” IEEE International Conference on Computer Vision (ICCV), pp. 2242-2251, 2017.
[36] Isola et al. “Image-to-Image Translation with Conditional Ad-versarial Networks” 2017 IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), pp. 5967-5976, 2016; Doi: 10.1109/CVPR.2017.632.
[37] Odena et al. “Conditional Image Synthesis with Auxiliary Classifier GANs” in International Conference on Machine Learning, 2016.
[38] R. Nagaraju and M. Stamp “Auxiliary-Classifier GAN for Malware Analysis” ArXiv, vol. abs/2107.01620, 2021.
[39] Donahue et al. “Adversarial Feature Learning” ArXiv, vol. abs/1605.09782, 2016.
[40] Xu et al. “Improved bidirectional GAN-based approach for network intrusion detection using one-class classifier” MDPI (Basel, Switzerland) 2022; Doi: 10.3390/computers 11060085.
[41] Saito et al. “Temporal Generative Adversarial Nets with Singular Value Clipping” in 2017. IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2849-2858; Doi: 10.1109/ICCV.2017.308.
[42] Munoz et al. “Multi-Variate Temporal GAN for Large Scale Video Generation” ArXiv, vol. abs/2004.01823, 2020.
[43] Mao et al. “Least Squares Generative Adversarial Networks” in 2017. IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2813-2821; Doi: 10.1109/ICCV.2017.304.
[44] Arjovsky et al. ”Wasserstein GAN,” ArXiv, vol. abs/1701.07875.
[45] Gulrajani et al. “Improved Training of Wasserstein GANs” in Neural Information Processing Systems, 2017.
[46] Wang et al. ”Evolutionary Generative Adversarial Networks” IEEE Transactions on Evolutionary Computation, vol. 23, pp. 921-934, 2019; Doi:10.1109/TEVC.2019.2895748
[47] Berthelot, et al. “BEGAN: Boundary Equilibrium Generative Adversar-ial Networks” ArXiv, vol. abs/1703. 10717, 2017.